Recurrent Neural Networks

Due at 4:00pm on 3 April 2018

What you need to get

- `YOU_a5.ipynb`: a Python notebook (hereafter called "the notebook")
- `origin_of_species.txt`: a text file

What to do

1. RNN

In this question, you will complete the Python implementation of backprop through time (BPTT) for a simple recurrent neural network (RNN). The notebook contains a definition for the class `RNN`. The class has a number of methods, including `BPTT`. However, `BPTT` is incomplete.

For training and testing, the notebook also reads in a corpus of text (a simplified version of *On the Origin of Species* by Charles Darwin), along with the character set, and creates about 5000 training samples. The notebook also creates a few utility functions that help convert between the various formats for the data.

(a) [8 marks] Implement the function `BPTT` so that it computes the gradients of the loss with respect to the connection weight matrices and the biases. Your code should work for different values of `seq_length` (this is the same as τ in the lecture notes).

(b) [2 marks] Create an instance of the RNN class. The hidden layer should have 400 ReLU neurons. The input to the network is a one-hot vector with 27 elements, one for each character in our character set. The output layer also has 27 neurons, with a softmax activation function.

(c) [2 marks] Train the RNN for about 15 epochs. Use cross entropy as a loss function (see A3 Q1 for help with this). You can use a learning rate of 0.001, but might want to break the training into 5-epoch segments, reducing the learning rate for each segment. Whatever works.

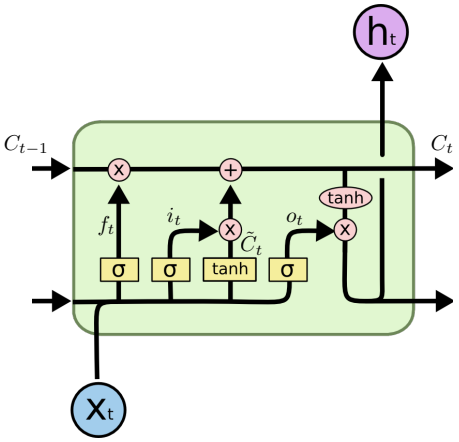
(d) [2 marks] What fraction of the time does your RNN correctly guess the first letter that follows the input? Write a small bit of Python code that counts how many times the next character is correct, and express your answer as a percentage in a print statement.

2. LSTM

The figure on the right shows a Long Short-Term Memory (LSTM) unit. At each step, it receives an input, x, a also recurrently recycles its hidden state, h, and the "cell" state, C. The formulas on the left detail the operation of the various gates.

Let $v_t = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}$ be the aggregated input.

$$
\begin{aligned}
f_t &= \sigma\left(W_f v_t + b_f\right) \\
i_t &= \sigma\left(W_i v_t + b_i\right) \\
o_t &= \sigma\left(W_o v_t + b_o\right) \\
\tilde{C}_t &= \tanh\left(W_C v_t + b_C\right) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
h_t &= o_t \odot \tanh\left(C_t\right)
\end{aligned}
$$



This figure was adapted from Colah's blog.

Let the connection weights be,

$$
\begin{aligned}
W_f &= [0\ 8\ 0\ 0] \quad b_f = -4 & \qquad W_i &= [0\ 0\ 9\ 0] \quad b_i = -4.5 \\
W_o &= [0\ 0\ 0\ 10] \quad b_o = -5 & \qquad W_C &= [1\ 0\ 0\ 0] \quad b_C = 0
\end{aligned}
$$

And the initial states be,

$$h_0 = [0.05] \qquad C_0 = [-0.02]$$

Note that $\frac{d}{d\phi}\tanh\phi \approx 1$ for small values of ϕ.

(a) [6 marks] Determine the outputs, C_t and h_t, for the inputs listed below. For each, show the values of the gates (f_t, i_t, o_t) and describe how the updated values, C_t and h_t, relate to the input values C_{t-1} and h_{t-1}.

i. $x_t = [1\ 0\ 0]$ ii. $x_t = [0\ 1\ 0]$ iii. $x_t = [1\ 0\ 1]$

(b) [2 marks] Suppose you want your new cell state, C_t, to approximate the sum of your old cell state and your hidden state, h_{t-1}. What should your input x_t be? Justify your answer, and demonstrate your solution on the LSTM setup given above.

(c) [3 marks] Suppose, instead, that you want your LSTM's output, h_t, to approximate the average of your previous cell state and the input value of h_{t-1}. What should your input x_t be? Justify your answer, and demonstrate your solution on the LSTM setup given above.

What to submit

Your assignment submission should be a single jupyter notebook file, named (`<WatIAM>_a5.ipynb`), where `<WatIAM>` is your UW WatIAM login ID (not your student number). The notebook must include solutions to **all** the questions. Submit this file to Desire2Learn. You do not need to submit any of the additional files supplied for the assignment.